

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-227402

(43)Date of publication of application : 24.08.2001

(51)Int.Cl.

F02D 45/00

G06F 11/10

G06F 12/16

(21)Application number : 2000-035229

(71)Applicant : DENSO CORP

(22)Date of filing : 14.02.2000

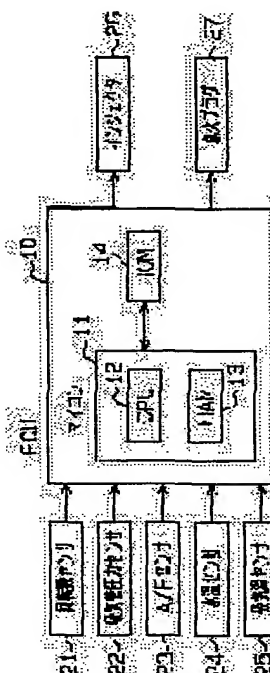
(72)Inventor : KONDO HIROSHI

## (54) ON-VEHICLE ELECTRONIC CONTROL DEVICE

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To efficiently operate a microcomputer and shorten a time till completion of calculation of a check sum.

**SOLUTION:** An ECU 10 is provided with a microcomputer 11 consisting of a CPU 12, RAM 13, etc., and a ROM 14 storing a control program, a data for deciding comparison, etc. The CPU 12 executes various engine controls such as fuel injection control and ignition timing control. The CPU 12 calculates a check sum for a specified address of the ROM 14. This check sum calculation is performed by time division, a number of addition bites is changed in accordance with a processing load of the CPU 12. For instance, a number of the addition bites is increased at low rotation time of an engine, in reverse, decreased at high rotation time of the engine.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]



Copyright (C); 1998,2003 Japan Patent Office



特開2001-227402

(2)

9)

【0003】ところで、車載電子制御装置（車載ECU）にはマイクロコンピュータが搭載され、該マイクロコンピュータ内のCPUにより各種のエンジン制御が実行され、エンジン制御が実行される。この場合、エンジン制御に加えてチェックサム算出が実施され、CPU処理負荷の増加をもたらす。それ従って、

・イグニッションキー（IGキー）のON直後におけるエンジン始動時、

・同IGキーのOFF直後におけるメインリレー制御時、

等々、エンジン制御によるCPUの処理負荷が比較的小さい状態でチェックサムが算出される。

【0004】上記の通りエンジン始動時にチェックサムを算出する場合、メモリ容量が比較的小さくチェックサム算出の所要時間が短ければチェックサムの算出完了までエンジン制御の開始を遅らせてもよい。しかしながら、各電子制御の高機能化に伴いメモリ容量が増大する傾向にあることを考えると、チェックサム算出の所要時間が増加し、その分、エンジン制御の開始時間も遅延される。従って、エンジン始動性が悪化することが懸念される。

【0005】また、メインリレー制御中にチェックサムを算出する場合、IGキーがOFFされるまでの期間はチェックサムが算出されず、メモリ容量があってもIGキーOFFまではそれが検出できないという不都合がある。

【0006】そこで近年では、上記の如くエンジン始動時にメインリレー制御時にチェックサムを算出する技術に代えて、通常のエンジン運転途中において、時々刻々チェックサムを算出する技術が提案されている。つまり、時々刻々でチェックサムを算出してCPUの処理負荷を分散させることにより、エンジン制御に並行してチェックサム算出を実施できるようにしている。

【0007】時々刻々でチェックサムを算出する場合において、エンジン回転数NEとCPU使用率との関係を図10に示す。なお、図10中、全CPU使用率のうち、エンジン制御によるCPU使用率を「A」で表し、チェックサム算出によるCPU使用率を「B」で表す。CPU使用率は、処理されるべきタスクが単位時間内に占める比率であり、タスクの全処理時間と単位時間とが一致する時、CPU使用率=100%であると定義される。【0008】図10（a）は、エンジン回転数NEの増大に伴い、エンジン制御の処理に占めるCPU使用率（図のA）は、エンジン回転数NEの増大に伴ってほぼ比例的に増えるのに対し、チェックサム算出によるCPU使用率（図のB）は、エンジン回転数NEにほぼ一定となる。この場合、全CPU使用率（図のA+B）が100%に達する回転数が処理境界回転数であり、全CPU使用率が100%を超えると、本

1

【特許請求の範囲】  
【請求項1】エンジン又は車載機器の各種制御に関する処理を定期的に実行するマイクロコンピュータを搭載し、該定期的に実行される処理毎にマイクロコンピュータがメモリ内のチェックサムを分割して算出する車載電子制御装置であり、  
マイクロコンピュータの処理負荷に応じて、チェックサム算出時に1度に算出するバイト数を変更することを特徴とする車載電子制御装置。

【請求項2】エンジン回転数をモニタする手段を設け、該回転数が高いほど、チェックサム算出時の加算バイト数を小さくする請求項1に記載の車載電子制御装置。

【請求項3】エンジン運転時にマイクロコンピュータがペーシング処理を繰り返して実行する際、該ペーシングの1回当たりの所要時間を計測し、その所要時間が大きいほど、チェックサム算出時の加算バイト数を小さくする請求項1に記載の車載電子制御装置。

【請求項4】エンジン回転数又は車速走行状態をモニタし、その状態がマイクロコンピュータの処理負荷を減じる状態にあれば、チェックサム算出時の加算バイト数を小さくする請求項1に記載の車載電子制御装置。

【請求項5】各種制御に関する複数の処理について、実行されるエンジン回転数領域が各々相違する車載電子制御装置であって、  
処理の重複/非重複が切り換えられるエンジン回転数に

応じて、チェックサムの算出時に1度に加算するバイト数を変更する請求項1に記載の車載電子制御装置。

【請求項6】エンジンの所定の回転数域では、チェックサムの算出を禁止する請求項1に記載の車載電子制御装置。  
【発明の詳細な説明】  
【0001】  
【発明の属する技術分野】本発明は、エンジン又は車載機器の各種制御を実行する機能と、メモリ内のデータを加算してチェックサム（メモリデータ加算値）を算出する機能とを併せ持つ車載電子制御装置に関する。

【従来の技術】従来より、メモリの異常検出や不正改変防止（ランパニング防止）を目的として、メモリのチェックサムを算出して車載電子制御装置が各種検出され、具体化されている。例えば、工場出荷時のメモリ検査等でメモリのチェックサムを算出し、そのチェックサムが算出されたメモリを判定してメモリの異常検出を行う技術が知られている。また、一般市場において、車両情報の一部としてメモリのチェックサムを算出し、そのチェックサムをメモリの不正改変を検出する技術が知られている。なお、今後は、メモリのチェックサム値を所定のダイアグナスタに出力することが法理化される可能性がある（ISO15031-5のmode S0

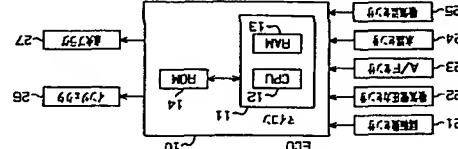
(11)特許出願公開番号  
特開2001-227402  
(P2001-227402A)  
(43)公開日 平成13年8月24日(2001.8.24)

(51)Int.Cl. <sup>7</sup>	F 02 D 45/00	F 1	チコト(参考)
F 02 D 45/00	372	F 02 D 45/00	372 F 3C 084
			372 C 5B 001
G 06 F 11/10	330	G 06 F 11/10	330 K 5B 018
12/16	320	12/16	320 B 9A 001

(21)出願番号	特開2000-35226(P2000-35226)	(71)出願人	00004060 株式会社社デンソー
(22)出願日	平成12年2月14日(2000.2.14)	(72)発明者	近藤 裕 愛知県刈谷市昭和町1丁目1番地 株式会社デンソー内
		(74)代理人	10068755 弁理士 豊田 博宣 (J41名) Fターム(参考) 320B R413 R415 R417 C409 D406 E402 E403 E406 F402 F411 F420 F426 F429 F433 5800 A414 D203 58018 G401 R413 J429 N404 R411 84001 R806 E302 U334

(54)【発明の名称】 車載電子制御装置

(57)【要約】  
【課題】マイクロコンピュータを効率的に動作させ、且つチェックサムの算出完了までの時間を短縮する。  
【解決手段】ECU10は、CPU12、RAM13等からなるマイコン11と、制御プログラムと比較判定用データ等を記憶したROM14とを備える。CPU12は、燃料噴射制御や点火時期制御といった各種エンジン制御を実行する。また、CPU12は、ROM14の指定されたアドレスについてチェックサムを算出する。このチェックサム算出は時々刻々で行い、CPU12の処理負荷に応じて加算バイト数を大きくし、逆に、エンジンの高回転時には加算バイト数を小さくする。





来実効すべきタスクが完全に実行できないこととなる。  
従って、全CPU使用率が100%を超す、エンジン制御等に悪影響を及ぼさない範囲で、チェックサム算出のために時分割する処理数、すなわち1度に加算する処理量(バイト数)を決定していた。

【0008】しかしながら、メモリが大容量化されつつある近年の車載電子制御装置の場合、上記時分割でチェックサムを算出手法では、チェックサム算出完了までの所要時間が長引くという問題が生じる。ここで、1度に加算するバイト数(時分割された1回の処理量)とチェックサムの算出完了時間とは、図11の図底にある、1度に加算するバイト数を増加させれば、算出完了時間が短縮できることが分かるがその反面、CPU処理時間が増加し、エンジン高回転域での処理限界を招く。すなわち、1度に加算するバイト数を増やした場合は、図10(b)に示すように、チェックサム算出によるCPU使用率(図のB)が増え、結果として処理限界回数を低下する。こうしたエンジン高回転域での処理限界を解消するにはCPUの処理能力を向上させることが考えられるが、この対策ではコスト増加を招く。

【0010】  
【表明が提供しようとする課題】本発明は、上記問題に着眼してなされたものであって、その目的とするところは、マイクロコンピュータを効率良く動作させ、且つチェックサムの算出完了までの時間を短縮することができ、車載電子制御装置を提供することである。

【0011】  
【課題を解決するための手段】請求項1に記載の車載電子制御装置では、マイクロコンピュータの処理負荷に応じた、チェックサム(サム値)の算出時に1度に加算するバイト数を変更するので、単位時間当たりのマイクロコンピュータの処理能力を考慮しつつチェックサム算出が実施されることとなる。つまり、マイクロコンピュータの処理負荷が比較的小さく、処理負荷の限界(使用率100%)に対して余裕があれば、チェックサム算出時の加算バイト数を大きくし、逆に、マイクロコンピュータの処理負荷が比較的大きく、処理負荷の限界に対して余裕がなければ、チェックサム算出時の加算バイト数を小さくする。これにより、マイクロコンピュータの実際の使用効率が増える。その結果、マイクロコンピュータを効率良く動作させ、且つチェックサムの算出完了までの時間を短縮することができる。

【0012】また本発明では、  
・請求項2に記載のように、エンジンの回転数をモニタする手段を設け、回転回数が高いほど、チェックサム算出時の加算バイト数を小さくする。  
・請求項3に記載したように、ベース期間の1回当りの所要時間を計算し、その所要時間が大きいほど、チェックサム算出時の加算バイト数を小さくする。  
といった構成を適宜採用すると良い。かかる場合、マイ

50 は、これらセンサ検出信号により検知されるエンジン回

クロコンピュータの処理負荷に応じた最適な加算バイト数が決定できる。なお、エンジン回転数が高いこと、ベース期間の1回当りの所要時間が大きいことは何れも、マイクロコンピュータの処理が過剰な状態であり、その処理負荷が増加することを意味する。

【0013】また、請求項4に記載したように、エンジン運転状態又は車両走行状態をモニタし、その状態がマイクロコンピュータの処理負荷を減じる状態であれば、チェックサム算出時の加算バイト数を大きくする。といった構成でも望ましい効果が得られる。なおここで言う「マイクロコンピュータの処理負荷を減じる状態」とは、燃料カット中、始動時判定中、エンスト判定中、アイドル判定中などを指す。

【0014】また、請求項5に記載の説明では、各種制御に関与する複数の処理について、実施されるエンジン回転数範囲が互に相違する車載電子制御装置であった。処理の実施/非実施が切り替えられるエンジン回転数に応じて、チェックサムの算出時に1度に加算するバイト数を変更する。かかる場合にも、マイクロコンピュータの処理負荷に応じた最適な加算バイト数が決定できる。

【0015】請求項6に記載の説明では、エンジンの所定の高回転域でチェックサムの算出を禁止することで、エンジン高回転域において、チェックサム算出以外に本来実施される処理(タスク)が実施され、あるいは実施が遅れるといった不都合が解消される。

【0016】  
【表明の実施形態】本実施形態では、本発明を車載エンジンの制御システムとして具体化し、車載電子制御装置(ECU)は、燃料噴射制御や点火時期制御等、各種のエンジン制御を実行する。また、ECUは、メモリの異常検出や不正改造検出を行うべく、メモリのデータのチェックサム(メモリデータ加算値)を算出する。以下、メモリのデータのチェックサムを詳細に説明する。

【0017】図1は、本実施形態におけるECU10の概略的構成を示すブロック図である。図1において、ECU10は、周知のマイクロコンピュータ(以下、マイコンという)11を中心に構成されており、同マイコン11は、各種制御プログラムを実行するCPU12、制御データ等を一時的に記憶するRAM13、その制御を示さないA/D変換器や出力ポート等を備える。また、ECU10は、制御プログラムや比較判定用データを格納したROM14を備える。

【0018】ECU10には、エンジン回転数を検出する回転数センサ21、エンジン吸気管内の圧力を検出する吸気管圧力センサ22、空燃比(A/F)を検出するA/Fセンサ23、エンジン冷却水の温度を検出する水温センサ24、吸気温度を検出する吸気温度センサ25等、各種センサより検出信号が入力される。ECU10

回転NEE、吸気管圧力PM、A/F、エンジン水温TW、吸気温度TA等に基づき、燃料噴射制御や点火時期制御等を実施する。すなわち、ECU10は、上記の各パラメータに基づいて燃料噴射制御のための噴射パルス信号を生成し、この噴射パルス信号によりインジェクタ8の噴射を制御する。また、ECU10は、同じく上記の各パラメータに基づいて点火制御のための点火信号を生成し、この点火信号により点火プラグ27の点火時期を制御する。

【0019】図2は、マイコン11内のCPU12が実行する制御プログラムの概略を示すフローチャートである。通常、プログラムは図2の(a)初期化処理と、(b)ベース処理と、(c)割り込み処理とから構成される。つまり、ECU10が起動されると、初期化処理→ベース処理の順に制御プログラムが起動され、その後、ベース処理が繰り返して実行される。また、回転数センサ21から等クランク角毎に回転パルス信号が入力されると、その駆動割り込み処理が要求され、これによりベース処理を中断して割り込み処理が実行される。割り込み処理終了後は、中断したベース処理に戻り、ベース処理が継続的に実行される。

【0020】詳しくは、図2(a)の初期化処理の起動されると、まずステップ110では、マイコン11の動作環境を設定する。そして、続くステップ120ではRAM13の初期化を行い、その後、図2(b)に示すベース処理→移行する。

【0021】ベース処理においてステップ130では、図示しない特定のマップを用い、その時のエンジン回転数NEEと吸気管圧力PMとに基づいて基本噴射量を算出する。続くステップ140では、エンジン水温TW、吸気温度TA等に基づいて噴射修正量を算出し、この噴射修正量により前記基本噴射量を補正する。またこのとき、A/Fと実A/Fとの偏差に基づいてフィードバック修正量を算出され、該修正量によるA/Fフィードバック制御も併せて実施される。この演算結果に基づいてインジェクタ28の噴射制御が制御される。更に、ステップ150では、エンジン回転数NEEや吸気管圧力PM等に基づいて点火信号を生成する。この点火信号により、点火プラグ27の点火時期が制御される。

【0022】その後、ステップ160では、ROMデータ中のチェックサムを算出する。つまり、ROM14内の規定されたアドレス領域についてデータを全て加算し、その和をチェックサム(サム値)とする。このチェックサムの算出値は、ROM14の異常検出や、ROM14の不正確な検出に用いられる。なお、チェックサム算出の詳細な手順については後述する。チェックサム算出後、ステップ130に戻り、以降の処理を繰り返して実行する。

【0023】また、図2(c)に示す割り込み処理は、例えばエンジンの回転に同期して起動される。本処理の

ステップ170では、回転数センサ21から入力される回転パルス信号を取り込み、その時々検出される時刻により本割り込みの周期を算出する。つまり、この割り込みの周期によりエンジン回転数が算出され、燃料噴射制御や点火時期制御に用いられる。なお図示は省略するが、割り込み処理には回転同期の割り込み以外に、マイコン11内部のタイマにより起動される定時割り込み等も含まれる。

【0024】次に、チェックサム算出の手順について図3のフローチャートを用いて説明する。図3において先ずステップ201では、チェックサムの加算開始アドレスを算出する(但し、初期化処理は図示していない)。すなわち、ROM14全体のチェックサム加算を時分割で実行するための開始アドレスを設定する。

【0025】続くステップ202では、1度に加算するバイト数(加算バイト数)を設定する。加算バイト数の決定処理の一例としては、図4に示すサブルーチンと呼ばれる処理を実行する。この図4の処理では、加算バイト数が18～128バイトの範囲内で可変に設定される。

【0026】図4について詳しくは、ステップ211では現在、燃料カット中、始動時判定中、エンスト判定中、アイドル判定中など、CPU負荷が比較的小さいエンジン運転状態であるかを判断する。燃料カット中など、CPU負荷が比較的小さい場合(ステップ211がYES)、加算バイト数が比較的大きくても良い。め、ステップ218に飛び、加算バイト数を128バイトとする。

【0027】また、ステップ211がNOの場合、ステップ212～218において、その時のエンジン回転数NEEに応じて加算バイト数を設定する。すなわち、CPU12は、エンジン回転数NEEが何れの回転域にあるかを判定し(ステップ213、215、217)、その回転域に応じて加算バイト数を決定する(ステップ212、214、216、218)。

【0028】この場合、エンジン回転数NEEと加算バイト数との関係を図示すれば、図5のようになる。図5によれば、CPU12を効率良く使用するための加算バイト数が回転域毎に決定され、チェックサム演算に伴うCPU使用率が最適化できることとなる。

【0029】加算バイト数が決定されると、図3に戻りステップ203では、ステップ201、202にて求められた割り込み加算終了アドレスを設定する。また、ステップ204では、実際にチェックサム加算処理を実行する。すなわち、ROM14内のアドレス1のデータを全て加算し、チェックサムを算出する。その後、ステップ205では、前記決定した加算終了アドレスに達したか否かを判断し、加算終了判定されるまでステップ204、205を繰り返す。加算終了アドレスに達すると、ステップ206では次の加算開始アドレスを設定し、本処理を終了する。

50 本処理を終了する。



【0030】図3の処理により、ROM14の加算開始アドレスから加算終了アドレスまでを一定分として、チェックサムが均等分割して算出され、その値は返して、メモリ、ROM全体のチェックサムが算出される。

【0031】図8には、時刻割によるCPU使用率とエンジン回転数NEとの関係を示す。なお図6中、全CPU使用率のうち、エンジン制御によるCPU使用率を「A」で表し、チェックサム算出によるCPU使用率を「B」で表す。

【0032】図6によれば、エンジン制御によるCPU使用率（図のA）は、エンジン回転数NEの増加に伴いはほぼ比例的に増える。つまり、エンジン高回転域では割り込み処理の頻度が増加し、CPU使用率が増加する。これに対し、チェックサム算出によるCPU使用率（図のB）は、エンジン回転数NEの増加に伴い段階的に減じられている。前記図10（a）の従来の技術と比較すると、図6では、エンジン制御によるCPU使用率が低い低回転域において、チェックサム算出によるCPU使用率が上がり、逆にエンジン制御によるCPU使用率が低い高回転域において、チェックサム算出によるCPU使用率が下がることが分かる。要するに、CPU使用率の増減が一定である。要するに、CPU使用率の増減が一定である。要するに、CPU使用率の増減が一定である。

【0033】図7は、所定の進行パターンにおけるチェックサム算出時間を示すタイムチャートである。図7において、（a）は進行パターン上のエンジン回転数NEの変化をモデル化して示し、（b）はNEに応じて可変に設定される加算バイト数を示し、（c）は前記（b）の加算バイト数を倍増した算出バイト数を示し、（d）は加算バイト数一定とする場合（従来技術）の算出バイト数を比較して示す。

【0034】（d）に示す従来技術の場合、エンジン回転数NEの定常値より一定量の加算バイト数が毎回算出され、時刻11でチェックサム算出が開始された後、時刻13でチェックサム算出が完了する。すなわち、チェックサム算出完了には「13-11」の時間を要する。

【0035】これに対し、本実施の形態の場合、（b）に示すようにエンジン回転数NEの変化に応じて1回の加算バイト数がその範囲で決定される。この場合、時刻11でチェックサム算出が開始された後、時刻12でチェックサム算出が完了する。すなわち、チェックサム算出完了には「12-11」の時間を要する。

【0036】つまり、従来技術では、エンジン高回転域でのCPU使用率の制限があるため、これを基準に加算バイト数が比較的小さく設定されていたが、本実施の形

態では、特に低回転域での加算バイト数が大きくなるため、1度に計算できる容量が増え、チェックサムの算出完了時間が短縮できる。

【0037】以上詳述した本実施の形態によれば、以下に示す効果が得られる。CPU12（マイコン11）の処理負荷に応じて加算バイト数を変更するので、単位時間当たりのCPU処理能力を考慮しつつチェックサム算出が実施され、CPU12の実質的な使用率が上がる。その結果、CPU12を効率的に動作させ、且つチェックサムの算出完了までの時間を短縮することができる。

【0038】この場合、ROM14が大容量化した場合でも、比較的短時間でチェックサム算出が完了できる。また、CPUの処理能力向上が図られることがないため、コストアップを招くこともない。

【0039】（第2の実施の形態）次に、本実施の形態で第2の実施の形態を説明する。但し、本実施の形態では、上述した第1の実施の形態と同等であるものは説明を省略し、第1の実施の形態との相違点を中心に説明する。

【0040】上記第1の実施の形態では、エンジン制御によるCPU使用率（使用率）がエンジン回転数NEに応じてほぼ比例的に上昇する旨を記載したが、実際のエンジン制御では、各種制御に関する複数の処理について、実施されるエンジン回転数範囲が重複する。つまり、一例として、低・中回転域（例えば0〜4000rpm）ではA/Fフィードバック制御が実施されるのに対し、高回転域（4000rpm以上）ではA/Fフィードバック制御が停止される。またその他、失火検出やアイドル回転制御の処理も低・中回転域でのみ実施される。

【0041】それ故、図8（a）に示すように、エンジン制御によるCPU使用率（図のA）がエンジン回転数NEに応じて一様に上昇するのではなく、変換点を持つことになる。

【0042】そこで本実施の形態では、上記図8（a）の図8に類似した加算バイト数を算出する。要するに、加算バイト数の算出に際し、前記図5の図8に代えて図9の図8（b）によれば、チェックサム算出によるCPU使用率（図のB）は、エンジン回転数NE並びに制御の変換点に応じて段階的に変化する。この場合にもやはり、回転数の全域でCPU使用率が向上するよう加算バイト数が最適化される。

【0043】こうして図9の図8に類似した加算バイト数を決定し、チェックサム算出を行う場合、チェックサム算出も各々が全CPU使用率は図8（b）のようになる。図8（b）によれば、チェックサム算出によるCPU使用率（図のB）は、エンジン回転数NE並びに制御の変換点に応じて段階的に変化する。この場合にもやはり、回転数の全域でCPU使用率が向上するよう加算バイト数が最適化される。

【0044】以上第2の実施の形態によれば、処理の実施/非実施が切り換えられるエンジン回転数（回転数）に応じたCPU処理負荷（変換点）に応じてチェックサム算出のための加算バイト数を算出するので、CPU処理負荷等に応じて加算バイト数を適正に決定することができ、図9に示すように、エンジン回転数NEと加算バイト数との関係が図8に比べて減少ではなく多様性を持たせることで、システム全体の通用性や汎用性が向上する。

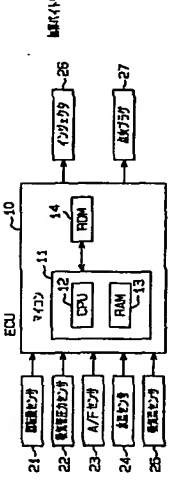
【0045】なお本発明は、上記以外に次の形態にて具現化できる。上記実施の形態では、概ねエンジン回転数NEに応じて加算バイト数を決定したが、これを以下の如く変更する。

（1）ベース処理の1個当たりの所要時間を計算し、その所要時間が大きいほど、チェックサム算出時の加算バイト数を小さくする。ベース処理の所要時間を計算する際は、例えば、フリーランカウンタを用い、図2（b）の0〜150〜180の間に1個当たりの所要時間を計算すれば良い。なおこの場合、ベース処理の所要時間が大きいことは、その途中に実施される割り込み処理等が増え、CPU処理負荷が増大することを意味する。

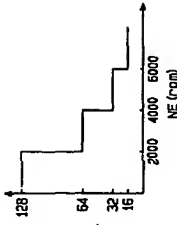
（2）単位時間内において、優先度が最も低い処理の実行回数を数え、その実施回数が多いほど、チェックサム算出時の加算バイト数を大きくする。なおこの場合、優先度が最も低い処理が実行されることは、単位時間内のCPU処理負荷が比較的低いことを意味する。

【0046】また、エンジン回転数の高回転域でチェックサムの算出を禁止する。例えば、前記図6に示す処理範囲回転数以上の回転域でチェックサムの算出を禁止する。要するに、所定の高回転域にある場合、チェックサム算出禁止フラグを立てる。或いは、前記図4の処理\*

【図1】



【図5】

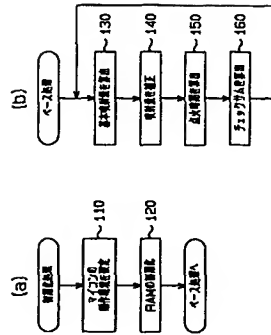




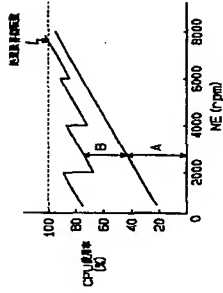
(3)

(7)

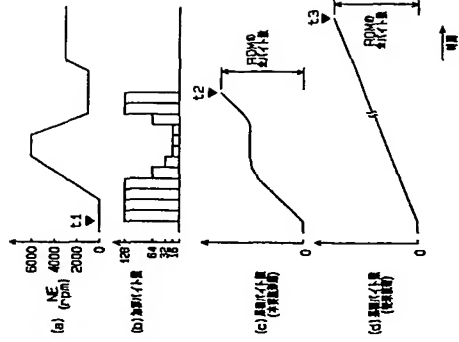
【図2】



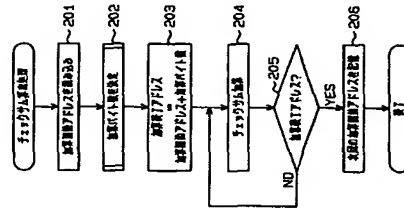
【図6】



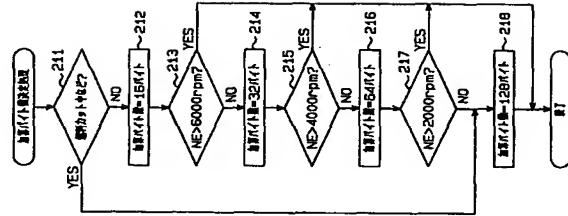
【図7】



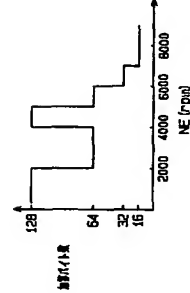
【図3】



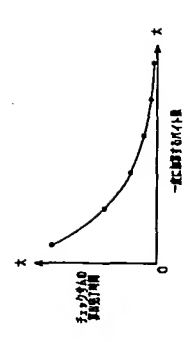
【図4】



【図9】

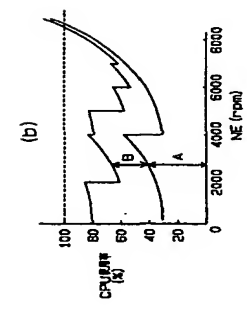
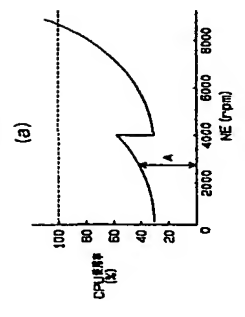


【図11】





(図8)



(図10)

